

**A Project Report**

**on**

**CRIME PREDICTION SYSTEM USING MACHINE LEARNING**

**Submitted in partial fulfilment of the requirements for the award of degree of**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE ENGINEERING**

**By**

**19WH1A0599      Ms. B. POOJITHA**

**19WH1A05A6      Ms. K. SAI PRANAVI**

**20WH5A0507      Ms. G. MANASWINI**

**Under the esteemed guidance of**

**Mr. C. Nagaraju**

**Assistant Professor**



**Department of Computer Science Engineering**

**BVRIT HYDERABAD**

**College of Engineering for Women**

**(NBA Accredited – EEE, ECE, CSE and IT)**

**(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)**

**Bachupally, Hyderabad – 500090**

**June-2023**

## **DECLARATION**

We hereby declare that the work described in this report, entitled “**CRIME PREDICTION SYSTEM USING MACHINE LEARNING** ” which is submitted by us in partial fulfillment for the award of the degree of Bachelor of Technology in the department of **Computer Science Engineering** at **BVRIT HYDERABAD College of Engineering for Women**, affiliated to **Jawaharlal Nehru Technological University Hyderabad**, Kukatpally, Hyderabad – 500085 is the result of original work carried out under the guidance of **Mr. C. Nagaraju, Assistant Professor, Department of CSE.**

**Ms. B. POOJITHA**

**(19WH1A0599)**

**Ms. K. SAI PRANAVI**

**(19WH1A05A6)**

**Ms. G. MANASWINI**

**(20WH5A0507)**

**Department of Computer Science Engineering**

**BVRIT HYDERABAD**

**College of Engineering for Women**

**(NBA Accredited – EEE, ECE, CSE and IT)**

**(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)**

**Bachupally, Hyderabad – 500090**



## **CERTIFICATE**

This is to certify that the project work report, entitled “**CRIME PREDICTION SYSTEM USING MACHINE LEARNING**” is a bonafide work carried out by **Ms. B.Poojitha (19WH1A0599)**, **Ms. K. Sai Pranavi (19WH1A05A6)**, **Ms. G. MANASWINI (20WH5A0507)** in partial fulfillment for the award of B.Tech degree in **Computer Science Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** under my guidance and supervision.

**Internal Guide**

**Mr. C. Nagaraju**

**Assistant Professor,**

**Department of CSE**

**Head of the Department**

**Dr. E. Venkateswara Reddy**

**Professor and HoD,**

**Department of CSE**

**External Examiner**

## **ACKNOWLEDGEMENTS**

We would like to express our sincere thanks to **Dr. K. V. N. Sunitha, Principal, BVRIT HYDERABAD College of Engineering for Women**, for her support by providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. E. Venkateshwara Reddy, Professor & HOD, Department of CSE, BVRIT HYDERABAD College of Engineering for Women**, for all timely support and valuable suggestions during the period of our project.

We are extremely thankful to our Internal Guide, **Mr. C. Nagaraju, Assistant Professor, CSE, BVRIT HYDERABAD College of Engineering for Women**, for her constant guidance and encouragement throughout the project.

Finally, we would like to thank our Major Project Coordinator, all Faculty and Staff of CSE department who helped us directly or indirectly. Last but not least, we wish to acknowledge our **Parents** and **Friends** for giving moral strength and constant encouragement.

**Ms. B. POOJITHA**

**(19WH1A0599)**

**Ms. K. SAI PRANAVI**

**(19WH1A05A6)**

**Ms. G. MANASWINI**

**(20WH5A0507)**

# LIST OF CONTENTS

| S.No | Topics                                       | Page No. |
|------|--|----------|
|      | Abstract                                     | V        |
|      | List of Figures                              | VI       |
| 1    | INTRODUCTION                                 | 1        |
|      | 1.1 Objective                                | 2        |
|      | 1.2 Methodology                              | 2        |
|      | 1.2.1 Dataset                                | 2        |
|      | 1.3 Organization of Project                  | 3        |
|      | 1.4 The Proposed Model                       | 4        |
| 2    | LITERATURE REVIEW                            | 5        |
| 3    | THEORETICAL ANALYSIS OF THE PROPOSED PROJECT | 11       |
|      | 3.1 Requirements Gathering                   | 11       |
|      | 3.1.1 Software Requirements                  | 11       |
|      | 3.1.2 Hardware Requirements                  | 11       |
|      | 3.2 Technological Descriptions               | 11       |
| 4    | DESIGN                                       | 14       |
|      | 4.1 Introduction                             | 14       |
|      | 4.2 Architecture                             | 15       |
|      | 4.3 Algorithms                               | 16       |
|      | 4.3.1 k-means clustering                     | 16       |
|      | 4.3.2 Random Forest                          | 23       |
| 5    | IMPLEMENTATION                               | 28       |
|      | 5.2 Evaluation Metrics                       | 28       |
|      | 5.2.1 RMSE                                   | 28       |
|      | 5.3 Results                                  | 29       |
| 6    | CONCLUSION AND FUTURE SCOPE                  | 32       |
| 7    | REFERENCES                                   | 33       |
| 8    | APPENDIX                                     | 34       |

## **ABSTRACT**

Prevention is better than Cure. Preventing a crime from occurring is better than investigating what or how the crime had occurred. Just like vaccination is given to a child to prevent disease, in today's world with such higher crime rate and brutal crime happenings, it has become necessary to have a vaccination system that prevents from crimes happening. By vaccinating society against crime, it refers to various methods such as educating peoples, creating awareness, increasing efficiency and proactive policing methods and other deterrent techniques. Crime prediction is of great significance to the formulation of policing strategies and the implementation of crime prevention and control. We are considering data of public crime from past years from a section of India as research data to assess the predictive power of machine learning algorithm. Results based on the historical crime data suggest that the clustering & regression performs well.

Data mining can play an important role in analyzing and predicting crimes using the data stored in repositories. Crime rate in India is increasing day by day, becoming a topic of major concern, hindering good governance in the world. Due to awful growth in crime rate, it has become impossible to analyze those crime related data and detect crime patterns or predict future crimes by intelligence agencies or local law enforcement agencies. This study will be helpful to law enforcing agencies in making strategies and tactics to address crime and disorder.

## List Of Figures

| S.NO | DESCRIPTION              | PAGE NO |
|------|--------------------------|---------|
| 1    | Architecture             | 15      |
| 2    | k-means algorithm        | 16      |
| 3    | Elbow method             | 18      |
| 4    | k-means clusters         | 19      |
| 5    | Random forest classifier | 25      |
| 6    | Random Forest regression | 26      |
| 7    | Root Mean Square Error   | 28      |
| 7    | Sample data set          | 29      |
| 8    | Input image              | 30      |
| 9    | Output image             | 30      |
| 10   | Clustering Output        | 31      |
| 11   | Analysis                 | 31      |

## 1. INTRODUCTION

The increase in crime data recording coupled with data analytics resulted in the growth of research approaches aimed at extracting knowledge from crime records to better understand criminal behavior and ultimately prevent future crimes. Crime is a complex social phenomenon that has grown due to major changes in society. Law enforcement agencies need to learn the factors that lead to an increase in crime tendency. To curb this, there is always a need for strategies and policies to prevent crime. Law enforcement agencies face a large volume of data that needs to be processed and turned into useful information.

Since crimes are increasing there is a need to solve the cases in a much faster way. The crime activities have been increased at a faster rate and it is the responsibility of police department to control and reduce the crime activities. Crime prediction and criminal identification are the major problems to the police department as there are tremendous amount of crime data that exist. There is a need of technology through which the case solving could be faster. Through many documentations and cases, it came out that machine learning and data science can make the work easier and faster. The aim of this project is to make crime prediction using the features present in the dataset. With the help of machine learning algorithm, using python as core we can predict the type of crime which will occur in an area. Visualization of dataset is done to analyze the crimes which may have occurred. This work helps the law enforcement agencies to predict and detect crimes with improved accuracy and thus reduces the crime rate.



## **1.1 Objective:**

The objective would be to develop a system which could perform analysis as well as prediction. Crime prediction & analysis are important to detect & understand future crimes. The main aim is to build the Crime Prediction System which will be a systematic approach for finding the future crime rate. Prediction and analysis are important to detect & understand future crimes. By building this, it can speed up the process of solving crimes. The ability to predict the crime which can occur in future can help the law enforcement agencies in preventing before it occurs.

## **1.2 Methodology:**

### **1.2.1 Data set:**

File Size: 11.2 KB Number of Rows: 837

Number of Columns: 14

Description: The dataset considered is an Indian dataset, the dataset contains crime data of different states. Each row represents a specific date, and the columns contain the following information:

“States”: This column specifies different states of India.

“Districts”: The Districts of various states of India.

“Year”: The year in which crime occurred in different states.

“Murder”: Murders occurred in different states in specific years.

“Rape”: Total number of rape rate.

“Kidnapping”: Total number of kidnapping rates.

“Robbery”: Total number of robbery rate in different states.

“Dowry”: Dowry cases occurred in different states in specific years.

“Dacoity”: Total number of Dacoity rate.

“Riots”: Total number of riots rate.

“Assault on Women”: Assault on women cases occurred in different states in specific years.

“Major Crime”: Total crime rate in different years in specific states.

### **1.3 Organization of Project**

The project crime rate prediction followed by analysis can be organized into several steps. Firstly, it is crucial to define the project's objectives clearly. This includes identifying the different crimes in specific years that occurred in different states and determining the future crime prediction. By establishing these goals, the project can maintain a clear direction throughout its execution.

The next step involves acquiring and preprocessing the necessary data. It is important to identify reliable sources for prediction that is specific states and years. This data will be used to train and validate the models. Preprocessing techniques should be applied to handle missing values, outliers, and normalize the data to ensure accurate and consistent results.

Feature engineering plays a vital role. Relevant features that are specified states and years need to be selected. Additionally, domain knowledge can be used to engineer additional features that provide valuable insights. These features will serve as inputs for the Clustering and Analysis.

After feature engineering, the appropriate models for Crime prediction and analysis need to be selected. K-Means is a popular choice for clustering data Analysis and can effectively differentiate the low and high crime rates. Random forest regression is used for future prediction of different crimes. Clustering and regression models are used for building analysis and prediction respectively.

Once the model selection is complete, the models need to be trained and validated. The dataset should be split into training, validation, and testing sets. The models are trained on the training set using crime data.

Clustering techniques to identify patterns, Clustering is a method to depict the dataset in the form of subsets called clusters so that the observations in the same cluster make some sense. It is a method of unsupervised learning and is used for statistical data analysis. For crime data clustering is applied to verify whether the particular crime is HIGH or LOW in the location.

This regression model predicts the Crime rate that is going to be happen in future by taking different parameters. Regression analysis can be described as a statistical technique used to predict/forecast values of a dependent variable given values of one or more independent variables. Once the models are trained and validated, evaluation can be performed. The models are applied to the testing set to simulate real-world scenarios, and their performance is assessed using various evaluation metrics such as RMSE, MSE, accuracy, precision, recall, F1-score, and profit/loss.

### **1.4 Proposed Model:**

The idea behind the project is to develop a model that predicts crime rate, thus giving us information on type of crime that may occur in future. By building this, it can speed up the process of solving crimes. K-means Algorithm & Random Forest regressor will be used in building up this system. Clustering algorithm is being used for crime prediction. The purpose of this work is to improve on previously proposed prediction framework through alternative crime mapping and provide an open-source implementation that police analysts can use to deploy more effective predictive policing.

Advantages:

- Good performance
- More precise & specific
- User friendly interface

## 2. LITERATURE SURVEY

**Title:** Crime Analysis Through Machine Learning

**Authors:** Suhong Kim, Param Joshi, Parminder Singh Kalsi, and Pooya Taheri

**Summary:**

In this paper, the author proposed machine-learning-based crime prediction. work includes Vancouver crime data for the last 15 years is analyzed using two different data-processing approaches. Machine-Learning predictive models, K-nearest-neighbor and boosted decision tree, are implemented and a crime prediction accuracy between 39% to 44% is obtained when predicting crime in Vancouver. The prediction accuracy can be improved by tuning both the algorithm and the data for specific applications. Although this model has low accuracy as a prediction model, it provides a preliminary framework for further analyses.

**Title:** Comparison of Machine Learning Algorithms for Predicting Crime Hotspots

**Authors:** XU ZHANG, LIN LIU LUZI XIAO, AND JIAKAI JI

**Summary:**

This paper takes the historical data of public property crime from 2015 to 2018 from a section of a large coastal city in the southeast of China as research data to assess the predictive power between several machine learning algorithms. In this paper, six machine learning algorithms are applied to predict the occurrence of crime hotspots in a town in the southeast coastal city of China. The following conclusions are drawn: 1. The prediction accuracies of LSTM model are better than those of the other models. It can better extract the pattern and regularity from historical crime data. 2. The addition of urban built environment covariates further improves the prediction accuracies of the LSTM model. It is found that the model with built environment covariates has better prediction effect compared with the original model that is based on historical crime data alone. Therefore, future crime prediction should take advantage of both historical crime data and covariates associated with criminological theories.

**Title:** An Exploration of Crime Prediction Using Data Mining on Open Data

**Authors:** Ginger Saltos Mihaela, Chen, and Yilong Yin

**Summary:**

While many of these approaches make use of clustering and association rule mining techniques, there are fewer approaches focusing on predictive models of crime. In this paper we explore models for predicting the frequency of several types of crimes by LSOA code (Lower Layer Super Output Areas – an administrative system of areas used by the UK police) and the frequency of anti-social behaviour crimes. Three algorithms are used from different categories of approaches: instance-based learning, regression and decision trees. The data are from the UK police and contain over 600,000 records before preprocessing. The results, looking at predictive performance as well as processing time, indicate that decision trees (M5P algorithm) can be used to reliably predict crime frequency in general, as well as antisocial behavior frequency.

**Title:** Risk terrain modeling: Brokering criminological theory and GIS methods for crime forecasting

**Authors:** J. M. Caplan, L. W. Kennedy, and J. Miller

**Summary:**

The research presented here has two key objectives. The first is to apply risk terrain modeling (RTM) to forecast the crime of shootings. The risk terrain maps that were produced from RTM use a range of contextual information relevant to the opportunity structure of shootings to estimate risks of future shootings as they are distributed throughout a geography. The second objective was to test the predictive power of the risk terrain maps over two six-month time periods, and to compare them against the predictive ability of retrospective hot spot maps. Results suggest that risk terrains provide a statistically significant forecast of future shootings across a range of cut points and are substantially more accurate than retrospective hot spot mapping. In addition, risk terrain maps produce information that can be operationalized by police administrators easily and efficiently, such as for directing police patrols to coalesced high-risk areas.

**Title:** Prediction of Crime Rate Analysis Using Supervised Classification  
Machine Learning Approach

**Authors:** Kirthika V, Padmanabhan A, Lavanya M, Lalitha S

**Summary:**

In this paper the analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score of decision tree algorithm/ Random Forest method. This brings some of the following insights about crime rate. It has become easy to find out relation and patterns among various data. It, mainly revolves around predicting the type of crime which may happen if we know the location of where it has occurred in real time world. Using the concept of machine learning we have built a model using training data set that have undergone data cleaning and data transformation. The model predicts the type of crime with accuracy of 80.

**Title:** Crime Prediction and Analysis

**Authors:** Pratibha, Akanksha Gahalot, Uprant, Suraina Dhiman, Lokesh Chouhan

**Summary:**

The prediction accuracy depends upon on type of data used, type of attributes selected for prediction. In, mobile network activity was used to obtain human behavioral data which was used to predict the crime hotspot in London with an accuracy of about 70% when predicting that whether a specific area in London city will be a hotspot for crime or not. Clustering approaches were used for detection of crime and classification method were used for the prediction of crime. On comparing the performance of different clustering algorithm DBSCAN gave result with highest accuracy and KNN classification algorithm is used for crime prediction. Hence, this system helps law enforcement agencies for accurate and improved crime analysis. In a comparison of classification algorithms, Naïve Bayes and decision tree was performed with a data mining software, WEKA. The datasets for this study were obtained from US Census 1990. In the pattern of road accidents in Ethiopia were studied after taking into consideration various factors like the driver, car, road conditions etc. Different classification algorithms used were K-Nearest Neighbor, Decision tree and Naive Bayes on a dataset containing around 18000 datapoints. The prediction accuracy for all three methods was between 69% to 71%

**Title:** Using geographically weighted regression to explore local crime patterns

**Authors:** M. Cahill and G. Mulligan

**Summary:**

In this research paper examines a structural model of violent crime in Portland, Oregon, exploring spatial patterns of both crime and its covariates. Using standard structural measures drawn from an opportunity framework, the study provides results from a global ordinary least squares model, assumed to fit for all locations within the study area. Geographically weighted regression (GWR) is then introduced as an alternative to such traditional approaches to modeling crime. The GWR procedure estimates a local model, producing a set of mappable parameter estimates and t-values of significance that vary over space. Several structural measures are found to have relationships with crime that vary significantly with location. Results indicate that a mixed model with both spatially varying and fixed parameters—may provide the most accurate model of crime. The present study demonstrates the utility of GWR for exploring local processes that drive crime levels and examining misspecification of a global model of urban violence.

**Title:** Language usage on Twitter predicts crime rates

**Authors:** A. Almeahadi, Z. Joudaki, and R. Jalali

**Summary:**

Social networks produce enormous quantity of data. Twitter, a microblogging network, consists of over 230 million active users posting over 500 million tweets every day. We propose to analyze public data from Twitter to predict crime rates. Crime rates have increased in the past recent years. Although crime stoppers are utilizing various technics to reduce crime rates, none of the previous approaches targeted utilizing the language usage (offensive vs. non-offensive) in Tweets as a source of information to predict crime rates. In this paper, we hypothesize that analyzing the language usage in tweets is a valid measure to predict crime rates in cities. Tweets were collected for a period of 3 months in the Houston and New York City by locking the collection by geographic longitude and latitude.

Further, tweets regarding crime events in the two cities were collected for verification of the validity of the prediction algorithm. We utilized Support Vector Machine (SVM) classifier to create a model of prediction of crime rates based on tweets. Finally, we report the validity of prediction algorithm in predicting crime rates in cities.

**Title:** Self-organised critical hot spots of criminal activity

**Authors:** H. Berestycki and J.-P. Nadal

**Summary:**

In this paper it has been introduced a family of models to describe the spatial-temporal dynamics of criminal activity. It is argued here that with a minimal set of mechanisms corresponding to elements that are basic in the study of crime, one can observe the formation of hot spots. By analyzing the simplest versions of our model, exhibited a self-organized critical state of illegal activities that we propose to call a warm spot or a tepid milieu depending on the context. It is characterized by a positive level of illegal or uncivil activity that maintains itself without exploding, in contrast with genuine hot spots where localized high level or peaks are being formed. Within our framework, we further investigate optimal policy issues under the constraint of limited resources in law enforcement and deterrence. We also introduce extensions of our model that take into account repeated victimization effects, local and long-range interactions, and briefly discuss some of the resulting effects such as hysteresis phenomena.

**Title:** A survey of data mining techniques for analyzing crime patterns

**Authors :** U. Thongsatapornwatana

**Summary:**

In recent years the data mining is data analyzing techniques that used to analyze crime data previously stored from various sources to find patterns and trends in crimes. In additional, it can be applied to increase efficiency in solving the crimes faster and also can be applied to automatically notify the crimes. However, there are many data mining techniques. In order to increase efficiency of crime detection, it is necessary to select the data mining techniques suitably. This paper reviews the literatures on various data mining applications, especially applications that applied to solve the crimes. Survey also throws light on research gaps and challenges of crime data mining.



In addition to that, this paper provides insight about the data mining for finding the patterns and trends in crime to be used appropriately and to be a help for beginners in the research of crime data mining.

**Title:** Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques

**Authors :** Wajiha Safat, Sohail Asghar, And Saira Andleeb Gillani.

**Summary:**

In recent years Despite considerable research efforts, yet there is a need to have a better predictive algorithm, which direct police patrols toward criminal activities. Previous studies are lacking to achieve crime forecasting and prediction accuracy based on learning models. Therefore, this study applied different machine learning algorithms, namely, the logistic regression, support vector machine (SVM), Naïve Bayes, k-nearest neighbors (KNN), decision tree, multilayer perceptron (MLP), random forest, and extreme Gradient Boosting (XGBoost), and time series analysis by long-short term memory (LSTM) and autoregressive integrated moving average (ARIMA) model to better the crime data. The performance of LSTM for time series analysis was reasonably adequate in order of magnitude of root mean square error (RMSE) and mean absolute error (MAE), on both data sets. Exploratory data analysis predicts more than 35 crime types and suggests a yearly decline in Chicago crime rate, and a slight increase in Los Angeles crime rate; with fewer crimes occurred in February as compared to other months. The overall crime rate in Chicago will continue to increase moderately in the future, with a probable decline in future years. The Los Angeles crime rate and crimes sharply declined, as suggested by the ARIMA model. Moreover, crime forecasting results were further identified in the main regions for both cities. Overall, these results provide early identification of crime, hot spots with higher crime rate, and future trends with improved predictive accuracy than with other methods and are useful for directing police practice and strategies.

### 3. THEORETICAL ANALYSIS OF THE PROPOSED PROJECT

#### 3.1 Requirements Gathering

##### 3.1.1 Software Requirements

Operating System: Windows 10

Programming Language: Python 3.8

Tool: Visual studio

Packages: NumPy, Pandas, Matplotlib, Scikit-learn

##### 3.1.2 Hardware Requirements

Processor: Intel core i5 processor

Memory: RAM 8GB

#### 3.2 Technological Description

**Python:** Python is a widely used programming language with excellent libraries and frameworks for data analysis, machine learning, and deep learning.

##### **Advantages of Python:**

- Presence of third-party modules
- Extensive support libraries (NumPy for numerical calculations, Pandas for data analytics, etc.)
- Open source and large active community base
- Versatile, Easy to read, learn and write
- High-level language
- Dynamically typed language (No need to mention data type based on the value assigned, it takes data type)
- Object-Oriented and Procedural Programming language
- Portable and Interactive
- Ideal for prototypes – provide more functionality with less coding
- Highly Efficient (Python's clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities.

- Internet of Things (IoT) Opportunities
- Interpreted Language
- Portable across Operating systems

**NumPy:** It is a library for the Python Programming Language which is used to support large multi-dimensional arrays and matrices along with a large collection of high-level mathematical functions to operate on the arrays. Since an image is basically a huge matrix of pixel values, NumPy is a great library for preprocessing the images.

**Pandas:** It is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze.

**Matplotlib:** It is a Plotting library for the Python Programming Language. It is a static library which is used to create static, animated and interactive visualizations. This project uses Matplotlib to verify the results of preprocessing.

**Scikit-learn (Sklearn):** is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistency interface in Python.

**Django:** is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

**Visual studio code:** Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.

It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich

ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET). It is used to develop computer programs including websites, web apps, web services and mobile apps.

**HTML:**HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

**CSS:** CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

## **4. DESIGN**

### **4.1 Design introduction**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the First step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software. The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage. During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

## 4.2 Architecture:

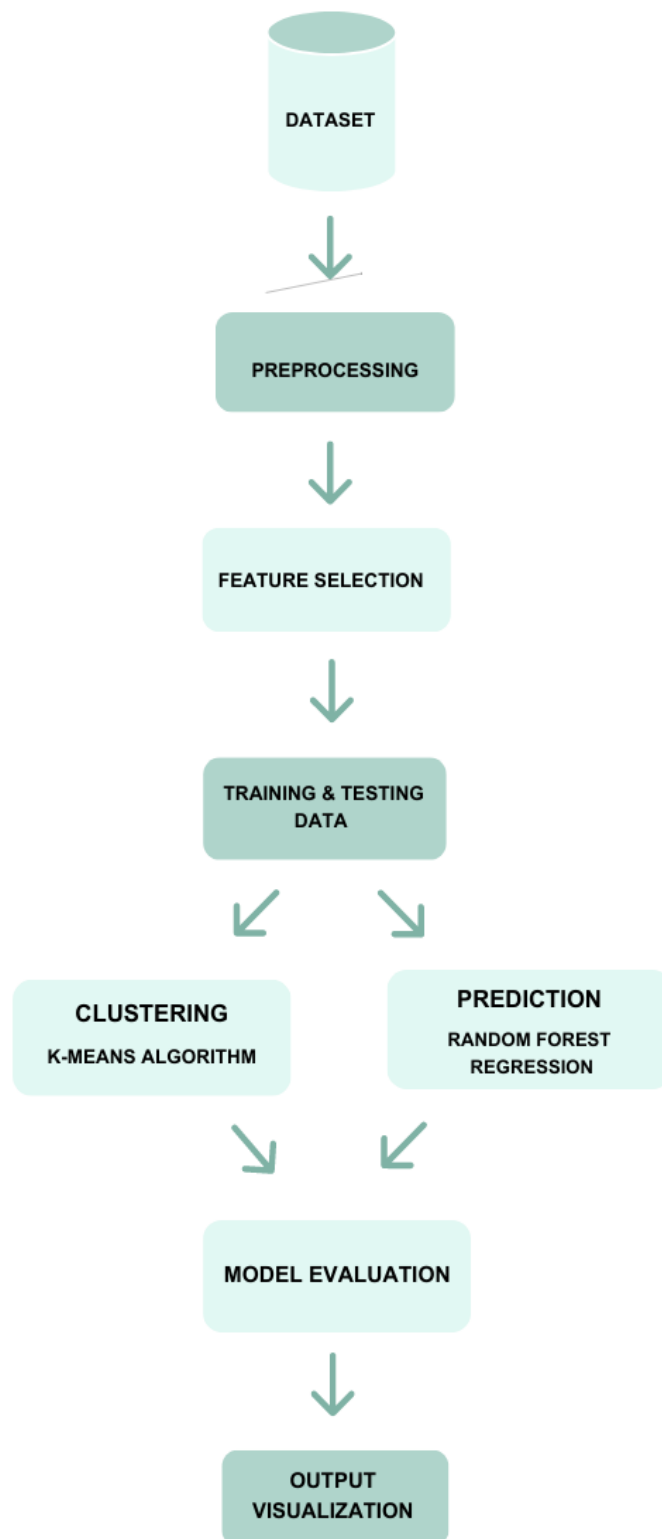


Fig. 1: Architecture

## 4.3 Algorithms

### 4.3.1 K-Means Clustering Algorithm

k-means is a technique for data clustering that may be used for unsupervised machine learning. It is capable of classifying unlabeled data into a predetermined number of clusters based on similarities (k).

The K-means clustering algorithm computes centroids and repeats until the optimal centroid is found. It is presumptively known how many clusters there are. It is also known as the flat clustering algorithm. The number of clusters found from data by the method is denoted by the letter 'K' in K-means.

In this method, data points are assigned to clusters in such a way that the sum of the squared distances between the data points and the centroid is as small as possible. It is essential to note that reduced diversity within clusters leads to more identical data points within the same cluster.

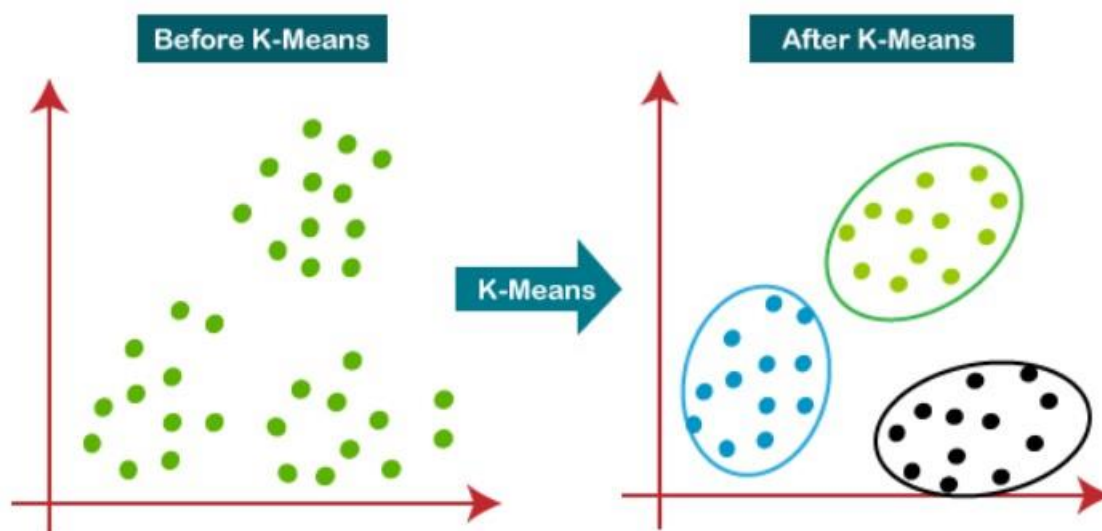


Figure 2: K-means Algorithm

K-means implements the Expectation-Maximization strategy to solve the problem. The Expectation-step is used to assign data points to the nearest cluster, and the Maximization-step is used to compute the centroid of each cluster.

When using the K-means algorithm, we must keep the following points in mind:

- It is suggested to normalize the data while dealing with clustering algorithms such as K-Means since such algorithms employ distance-based measurement to identify the similarity between data points.
- Because of the iterative nature of K-Means and the random initialization of centroids, K-Means may become stuck in a local optimum and fail to converge to the global optimum. As a result, it is advised to employ distinct centroids' initializations.

### **How to choose the k value?**

The end result of the algorithm depends on the number of clusters (k) that's selected before running the algorithm. However, choosing the right k can be hard, with options varying based on the dataset and the user's desired clustering resolution.

The smaller the clusters, the more homogeneous data there is in each cluster. Increasing the k value leads to a reduced error rate in the resulting clustering. However, a big k can also lead to more calculation and model complexity. So, we need to strike a balance between too many clusters and too few. The most popular heuristic for this is the elbow method. Below you can see a graphical representation of the elbow method. We calculate the variance explained by different k values while looking for an "elbow" – a value after which higher k values do not influence the results significantly. This will be the best k value to use.



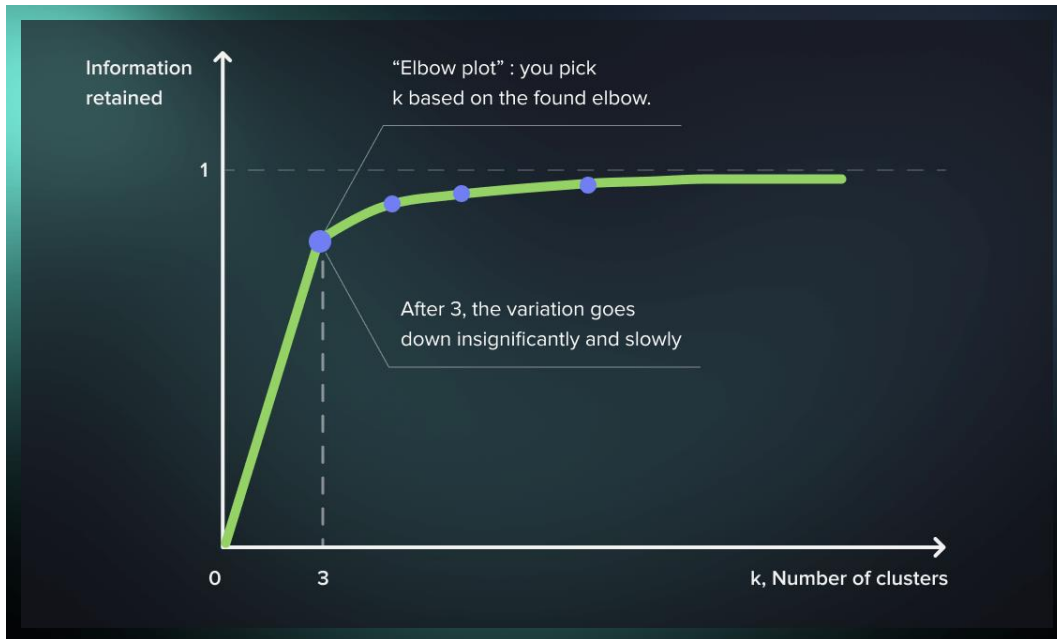


Fig.3 Elbow method

Most commonly, Within Cluster Sum of Squares (WCSS) is used as the metric for explained variance in the elbow method. It calculates the sum of squares of distance from each centroid to each point in that centroid's cluster. We calculate this metric for the range of  $k$  values we want to test and look at the results. At the point where WCSS stops dropping significantly with each new cluster added, adding another cluster to the model won't really increase its explanatory power by a lot. This is the "elbow," and we can safely pick this number as the  $k$  value for the algorithm.

### Implementation of K Means Clustering Graphical Form

STEP 1: Let us pick  $k$  clusters, i.e.,  $K=2$ , to separate the dataset and assign it to its appropriate clusters. We will select two random places to function as the cluster's centroid.

STEP 2: Now, each data point will be assigned to a scatter plot depending on its distance from the nearest  $K$ -point or centroid. This will be accomplished by establishing a median between both centroids.

STEP 3: The points on the line's left side are close to the blue centroid, while the points on the line's right side are close to the yellow centroid. The left Form cluster has a blue centroid, whereas the right Form cluster has a yellow centroid.

STEP 4: Repeat the procedure, this time selecting a different centroid. To choose the new centroids, we will determine their new center of gravity, which is represented below:

STEP 5: After that, we'll re-assign each data point to its new centroid. We shall repeat the procedure outlined before (using a median line). The blue cluster will contain the yellow data point on the blue side of the median line.

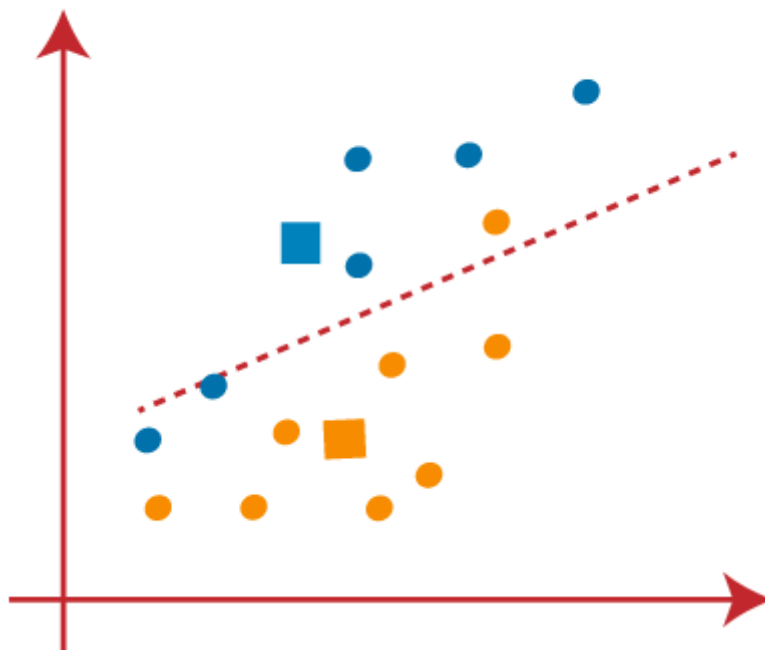
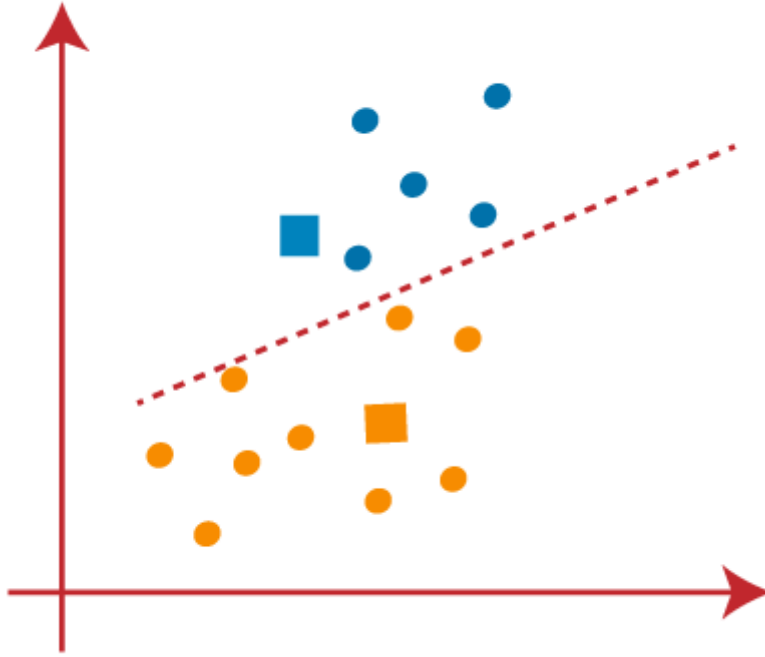


Fig.4 k-means clusters

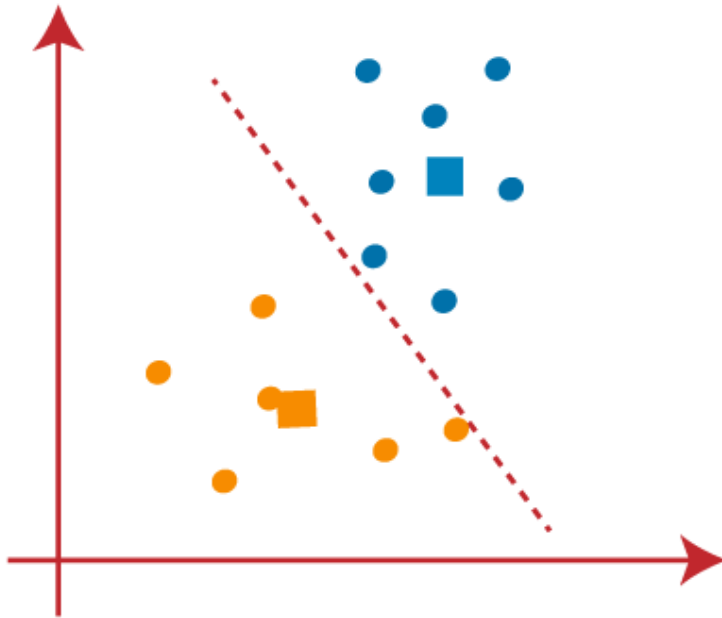
STEP 6: Now that reassignment has occurred, we will repeat the previous step of locating new centroids.



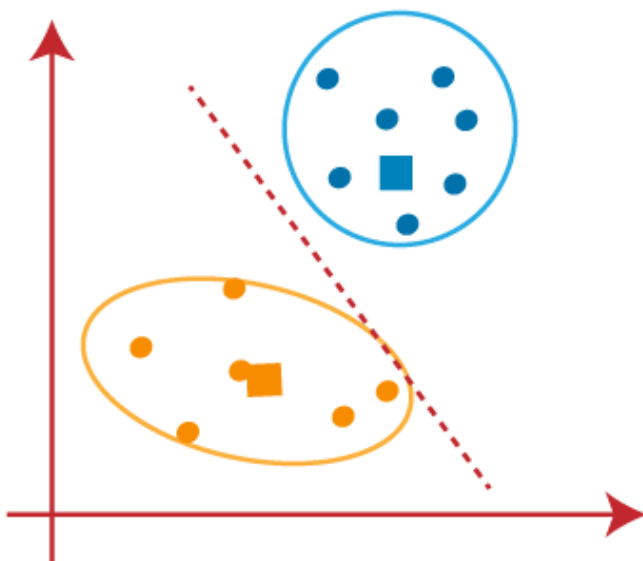
STEP 7: We will repeat the procedure outlined above for determining the centre of gravity of centroids, as shown below.



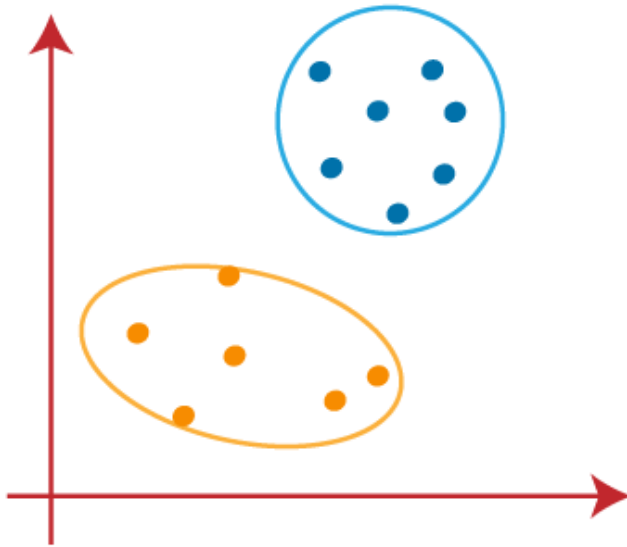
STEP 8: Similar to the previous stages, we will draw the median line and reassign the data points after locating the new centroids.



STEP 9: We will finally group points depending on their distance from the median line, ensuring that two distinct groups are established and that no dissimilar points are included in a single group.



The final Cluster is as follows:



Advantages:

- It is simple to grasp and put into practice.
- K-means would be faster than Hierarchical clustering if we had a high number of variables.
- An instance's cluster can be changed when centroids are re-computation.
- When compared to Hierarchical clustering, K-means produces tighter clusters.

Disadvantages:

- A major effect on output is exerted by initial inputs such as the number of clusters in a network (value of k).
- The sequence in which the data is entered has a considerable impact on the final output.
- It's quite sensitive to rescaling. If we rescale our data using normalization or standards, the outcome will be drastically different. ultimate result
- It is not advisable to do clustering tasks if the clusters have a sophisticated geometric shape.

### 4.3.2 Random Forest Algorithm

Random Forest is one of the most popular and commonly used algorithms by Data Scientists. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. The algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

### **Working of Random Forest Algorithm**

Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble learning technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make

predictions rather than an individual model.

Ensemble uses two types of methods:

1. Bagging— It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
2. Boosting— It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

### **Steps Involved in Random Forest Algorithm:**

Step 1: In the Random Forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.

### Classification in random forest:

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.

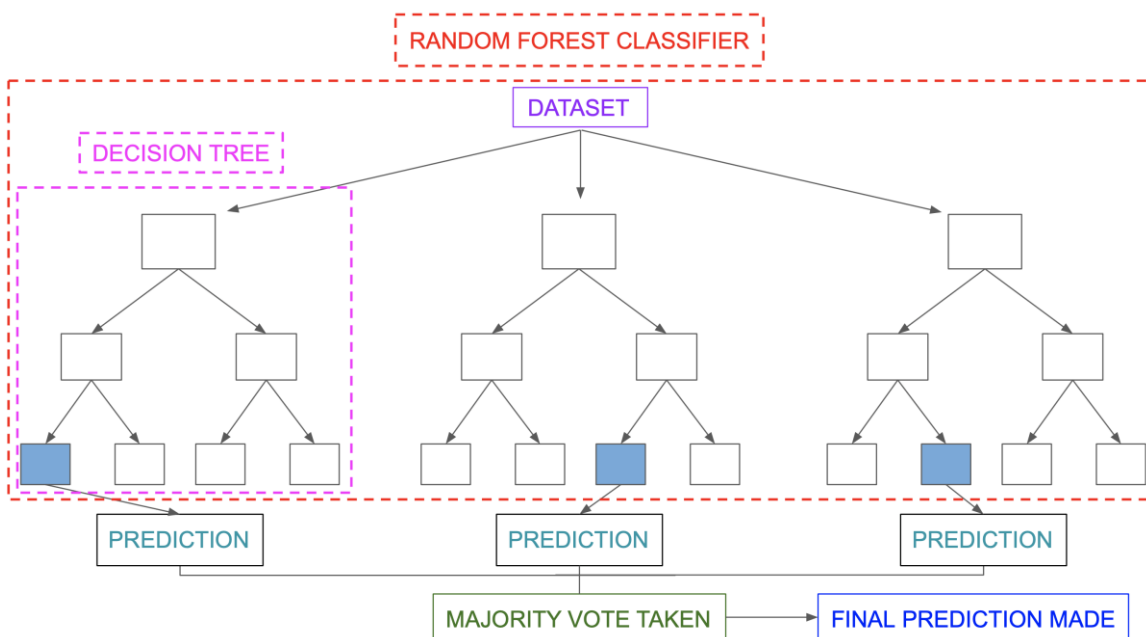


Fig.5 random forest classifier



### Regression in random forest:

Regression is the other task performed by a random forest algorithm. A random forest regression follows the concept of simple regression. Values of dependent (features) and independent variables are passed in the random forest model.

We can run random forest regressions in various programs such as SAS, R, and python. In a random forest regression, each tree produces a specific prediction. The mean prediction of the individual trees is the output of the regression. This is contrary to random forest classification, whose output is determined by the mode of the decision trees' class. Although random forest regression and linear regression follow the same concept, they differ in terms of functions. The function of linear regression is  $y = bx + c$ , where  $y$  is the dependent variable,  $x$  is the independent variable,  $b$  is the estimation parameter, and  $c$  is a constant.

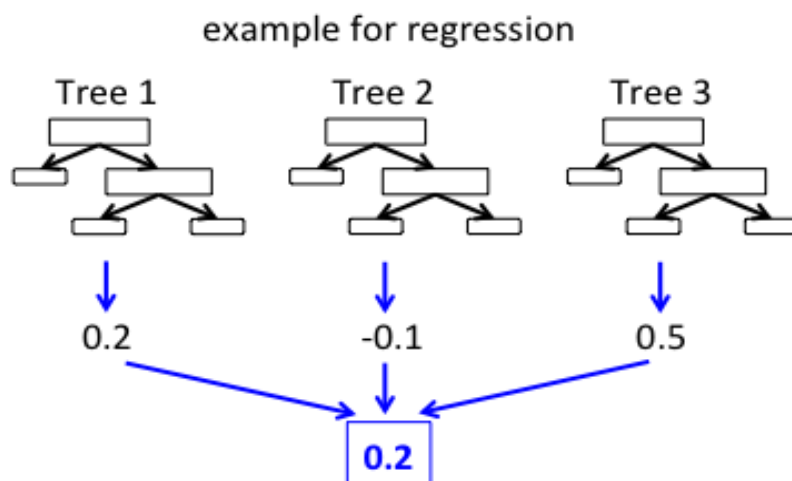


Fig.6 Random Forest regression

### Advantages:

- It can be used in classification and regression problems.
- It solves the problem of overfitting as output is based on majority voting or averaging.
- It performs well even if the data contains null/missing values.
- Each decision tree created is independent of the other; thus, it shows the property of parallelization.
- It is highly stable as the average answers given by a large number of trees are taken.
- It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases.
- It is immune to the curse of dimensionality. Since each tree does not consider all the attributes, feature space is reduced.

### Disadvantages:

- Random forest is highly complex compared to decision trees, where decisions can be made by following the path of the tree.
- Training time is more than other models due to its complexity. Whenever it has to make a prediction, each decision tree has to generate output for the given input data.

## 5. IMPLEMENTATION

### 5.1. Evaluation metrics

#### 5.1.1. Root Mean Square Error

Root-Mean-Square-Error or RMSE is one of the most popular measures to estimate the accuracy of our forecasting model's predicted values versus the actual or observed values while training the regression models or time series models. It measures the error in our predicted values when the target or response variable is a continuous number.

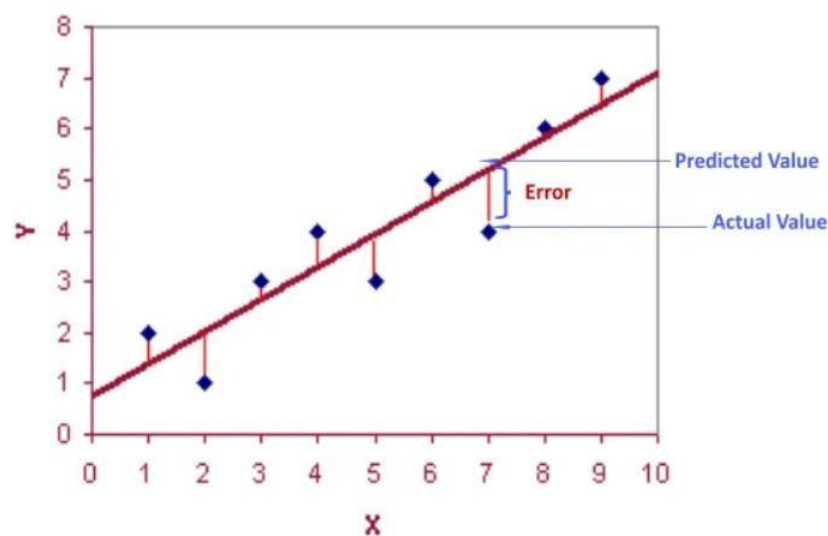


Fig. 7. Root Mean Squared Error

Thus, RMSE is a standard deviation of prediction errors or residuals. It indicates how spread out the data is around the line of best fit. It is also an essential criterion in shortlisting the best performing model among different forecasting models that you may have trained on one particular dataset. To do so, simply compare the RMSE values across all models and select the one with the lowest value on RMSE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - o_i)^2}$$

Where:

- $\sum$  is the summation of all values
- $f$  is the predicted value
- $o$  is observed or actual value
- $(f_i - o_i)^2$  are the differences between predicted and observed values and squared
- $N$  is the total sample size

## 5.2.2 Results:

|    | A  | B                    | C    | D      | E    | F         | G       | H       | I     | J     | K        | L         | M            | N | O |
|----|----|----------------------|------|--------|------|-----------|---------|---------|-------|-------|----------|-----------|--------------|---|---|
| 1  | Id | States/UT: District  | Year | Murder | Rape | Kidnappin | Dacoity | Robbery | Theft | Riots | Dowry_De | Assault_o | major_crimes |   |   |
| 2  | 0  | Andhra Pr Anantapu   | 2014 | 134    | 35   | 0         | 6       | 30      | 753   | 214   | 25       | 436       | 8376         |   |   |
| 3  | 1  | Andhra Pr Anantapu   | 2015 | 84     | 32   | 4         | 12      | 22      | 528   | 134   | 17       | 135       | 5374         |   |   |
| 4  | 2  | Andhra Pr Anantapu   | 2016 | 80     | 28   | 0         | 3       | 16      | 638   | 104   | 16       | 215       | 5803         |   |   |
| 5  | 3  | Andhra Pr Anantapu   | 2017 | 64     | 85   | 0         | 3       | 24      | 903   | 27    | 7        | 519       | 7630         |   |   |
| 6  | 4  | Andhra Pr Anantapu   | 2018 | 14     | 0    | 0         | 2       | 4       | 413   | 1     | 0        | 0         | 490          |   |   |
| 7  | 5  | Andhra Pr Guntur     | 2016 | 105    | 49   | 12        | 5       | 24      | 711   | 78    | 16       | 245       | 6897         |   |   |
| 8  | 6  | Andhra Pr Guntur     | 2017 | 51     | 40   | 0         | 5       | 31      | 1045  | 8     | 12       | 160       | 5798         |   |   |
| 9  | 7  | Andhra Pr Guntur     | 2018 | 51     | 80   | 20        | 0       | 15      | 521   | 9     | 8        | 354       | 7078         |   |   |
| 10 | 8  | Andhra Pr Kurnool    | 2014 | 118    | 32   | 4         | 3       | 28      | 563   | 108   | 12       | 403       | 8398         |   |   |
| 11 | 9  | Andhra Pr Kurnool    | 2018 | 78     | 58   | 71        | 8       | 44      | 1058  | 84    | 14       | 295       | 8175         |   |   |
| 12 | 10 | Andhra Pr Prakashar  | 2014 | 75     | 50   | 17        | 5       | 23      | 617   | 124   | 8        | 197       | 6367         |   |   |
| 13 | 11 | Andhra Pr Prakashar  | 2015 | 18     | 35   | 5         | 3       | 14      | 361   | 16    | 3        | 121       | 2430         |   |   |
| 14 | 12 | Andhra Pr Prakashar  | 2016 | 30     | 40   | 26        | 0       | 8       | 236   | 17    | 3        | 214       | 4727         |   |   |
| 15 | 13 | Andhra Pr Prakashar  | 2017 | 35     | 18   | 3         | 2       | 8       | 1148  | 31    | 13       | 60        | 3651         |   |   |
| 16 | 14 | Andhra Pr Vijayawac  | 2018 | 23     | 64   | 13        | 3       | 48      | 2016  | 2     | 10       | 279       | 7876         |   |   |
| 17 | 15 | Andhra Pr Vijayawac  | 2018 | 1      | 0    | 3         | 0       | 7       | 1085  | 0     | 2        | 8         | 1223         |   |   |
| 18 | 16 | Andhra Pr Visakha Ri | 2015 | 43     | 38   | 18        | 5       | 5       | 179   | 14    | 4        | 129       | 3028         |   |   |
| 19 | 17 | Andhra Pr Visakha Ri | 2016 | 38     | 84   | 130       | 4       | 46      | 1358  | 3     | 13       | 147       | 7728         |   |   |
| 20 | 18 | Andhra Pr Visakha Ri | 2017 | 43     | 47   | 0         | 2       | 4       | 388   | 13    | 8        | 225       | 5013         |   |   |
| 21 | 19 | Andhra Pr Visakha Ri | 2018 | 90     | 146  | 29        | 4       | 32      | 1096  | 26    | 24       | 405       | 8542         |   |   |
| 22 | 20 | Andhra Pr Total      | 2015 | 1175   | 961  | 355       | 75      | 433     | 15617 | 1013  | 215      | 4547      | 114604       |   |   |
| 23 | 21 | Arunachal Anjaw      | 2015 | 0      | 0    | 0         | 0       | 0       | 2     | 1     | 0        | 1         | 29           |   |   |
| 24 | 22 | Arunachal Changlang  | 2015 | 9      | 6    | 0         | 1       | 5       | 50    | 1     | 0        | 0         | 180          |   |   |

Fig.8 Data set

## Crime Prediction System Using Machine Learning

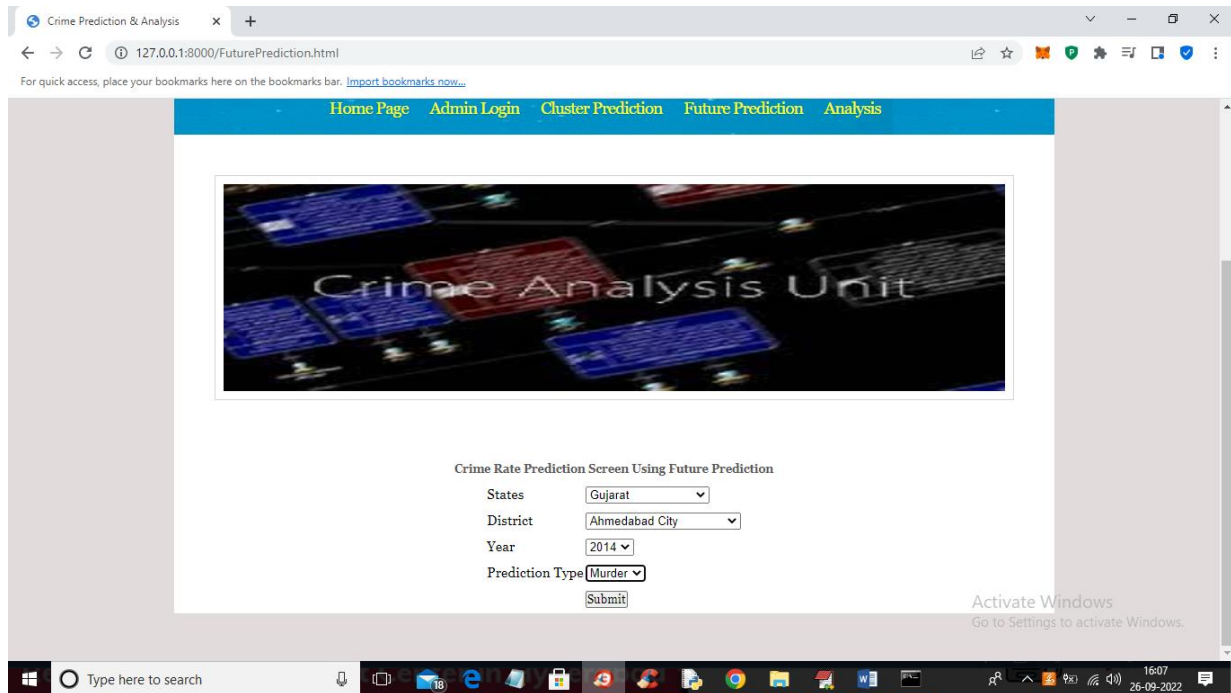


Fig.9 Input image

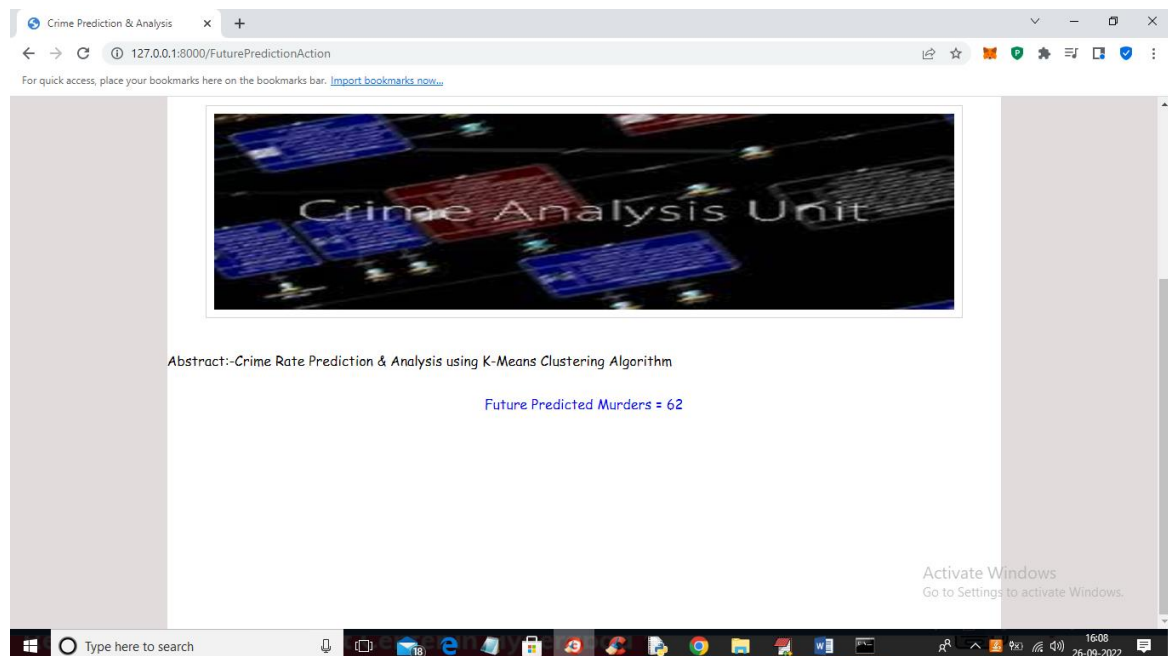



Fig.10 Output image



Crime Rate Prediction Screen Using Clustering

States

District

Year

Murder

Rape

Theft

Dowry Deaths

Anantapur Low Crime Rate Area

Fig:11 Clustering Output

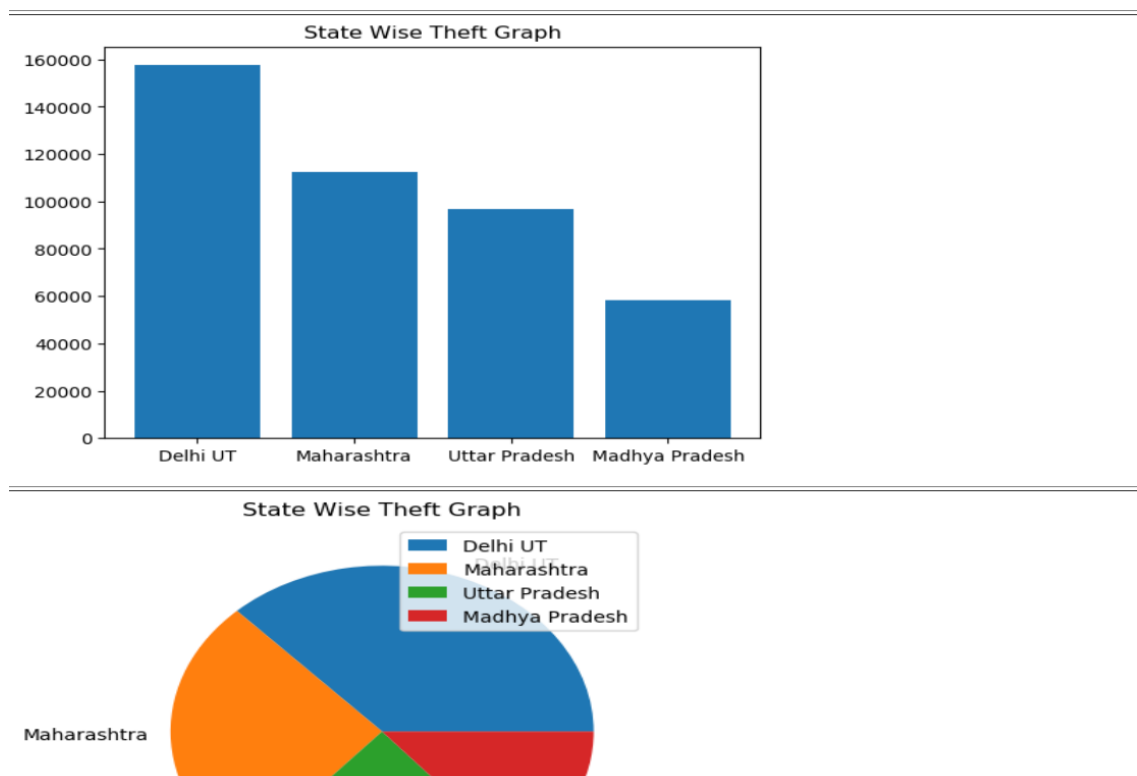


Fig.12 Analysis

## 6. CONCLUSION AND FUTURE SCOPE

The process started from data cleaning and processing, missing value and finally model building and evaluation. This brings some of the following insights about crime rate. It has become easy to find out relation and patterns among various data. It, mainly revolves around predicting the type of crime which may happen if we know the location of where it has occurred in real time world. Using the concept of machine learning we have built a model using training data set that have undergone data cleaning and data transformation. In this project we are using clustering and regression algorithms to predict crime rate. Clustering algorithm is used to predict HIGH or LOW crime area and regression algorithms are used to forecast future crime rate. Visualization of dataset is done to analyze the crimes which gives better understanding visually.

There is a lack of proper standards across the globe to record the crimes that happen. Even if the data is available, it is not easy to accommodate that data with another framework to achieve more concrete predictions. These are challenges for crime analysis to handle. These should be handled in future research. As of now, the project will rely on manual input from a human (a police officer) in order to enter details in the database. If we can make this a centralized system and connect it to all the police stations countrywide and make FIR reporting digital, then it would be quite easier to predict crimes in that particular location and recognize patterns in them.

## 7. REFERENCES

**Base Paper** – XU ZHANG, LIN LIU, LUZI XIA, AND JIAKAI JI, “Comparison of Machine Learning Algorithms for Predicting Crime Hotspots”, IEEE ACCESS,2020.

[1] Akanksha Gahalot,Uprant,Suraina Dhiman,”Crime Prediction and Analysis”,(IEEE) Conference Paper-February 2021

[2] Akash,Aniket verma,Nidi Lal,Yash,”Crime Prediction Using K-Nearest Neighbouring Algorithm”, International,conference on emerging trends in information technology and engineering-2020

[3] Ch. Mahendra,G. Nani Babu,G. Balu Nitin Chandra,A. Avinash,”CRIME RATE PREDICTION”,journal of engineering stories(JES), Vol 11, Issue 5,May/2020

[4] Kirthika V, Krithika Padmanabhan A , Lavanya,”Prediction of Crime Rate Analysis Using Supervised Classification Machine Learning Approach”, (IRJET) Volume: 06| Mar 2019

[5] Suhong Kim, Param Joshi, Parminder Singh Kalsi,”Crime Analysis Through Machine Learning”, Fraser International College, Simon Fraser University Conference Paper-2018

[6] Varshitha D N, Aishwarya P, Sahana R, “Paper on Different Approaches for Crime Prediction system”, (IJERT), NCETEIT - 2017 Conference Proceedings

[7] H. Berestycki and J.-P. Nadal,” Self-organised critical hot spots of criminal activity” IEEE Access (Volume: 9), 03 February-2018

[8] A. Almeahmadi, Z. Joudaki, and R. Jalali,”Language usage on Twitter predicts crime rates”



## 8. APPENDIX

```
import matplotlib.pyplot as plt

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

import os

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import MinMaxScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import train_test_split

from sklearn.cluster import KMeans

from sklearn.metrics import mean_squared_error

from sklearn.model_selection import train_test_split

from math import sqrt

global dataset, kmeans_cluster, theft_cls, rape_cls, murder_cls
```

```
sc = MinMaxScaler(feature_range = (0, 1))

le1 = LabelEncoder()

le2 = LabelEncoder()


le3 = LabelEncoder()

le4 = LabelEncoder()

global mse, rmse

def calculateError(alg, X_test, y_test):

    predict = alg.predict(X_test)

    #predict = predict.reshape(predict.shape[0],1)

    #predict = sc.inverse_transform(predict)

    predict = predict.ravel()

    #labels = sc.inverse_transform(y_test)

    labels = y_test.ravel()

    mse_error = mean_squared_error(labels,predict)

    rmse_error = sqrt(mse_error)

    mse.append(mse_error/1000)

    rmse.append(rmse_error)


def UploadDatasetAction(request):

    if request.method == 'POST':
```

## Crime Prediction System Using Machine Learning

global dataset, kmeans\_cluster, theft\_cls, rape\_cls, murder\_cls, mse, rmse

mse = []

rmse = []

myfile = request.FILES['t1']

```
dataset = pd.read_csv("Dataset/Dataset.csv", usecols=['States/UTs','District',
'Murder', 'Rape', 'Theft', 'Dowry_Deaths', 'Year'])
```

```
dataset.fillna(0, inplace = True)
```

```
cols = ['States/UTs', 'District']
```

```
dataset[cols[0]] = pd.Series(le1.fit_transform(dataset[cols[0]].astype(str)))
```

```
dataset[cols[1]] = pd.Series(le2.fit_transform(dataset[cols[1]].astype(str)))
```

```
X = dataset.values
```

```
X = sc.fit_transform(X)
```

```
kmeans_cluster = KMeans(n_clusters=2, n_init=1200)
```

```
kmeans_cluster.fit(X)
```

```
dataset = pd.read_csv("Dataset/Dataset.csv", usecols=['States/UTs','District', 'Year',
'Theft', 'Murder', 'Rape'])
```

```
dataset.fillna(0, inplace = True)
```

```
print(dataset)
```

```
cols = ['States/UTs', 'District']
```

```
dataset[cols[0]] = pd.Series(le3.fit_transform(dataset[cols[0]].astype(str)))
```

```
dataset[cols[1]] = pd.Series(le4.fit_transform(dataset[cols[1]].astype(str)))
```

```
theft_Y = dataset['Theft'].values
```

## Crime Prediction System Using Machine Learning

```
murder_Y = dataset['Murder'].values
```

```
rape_Y = dataset['Rape'].values
```

```
dataset.drop(['Theft'], axis = 1,inplace=True)
```

```
dataset.drop(['Murder'], axis = 1,inplace=True)
```

```
dataset.drop(['Rape'], axis = 1,inplace=True)
```

```
X = dataset.values
```

```
X_train1, X_test1, y_train1, y_test1 = train_test_split(X, theft_Y, test_size = 0.2)
```

```
X_train2, X_test2, y_train2, y_test2 = train_test_split(X, rape_Y, test_size = 0.2)
```

```
X_train3, X_test3, y_train3, y_test3 = train_test_split(X, murder_Y, test_size = 0.2)
```

```
theft_cls = RandomForestRegressor()
```

```
theft_cls.fit(X, theft_Y)
```

```
calculateError(theft_cls, X_test1, y_test1)
```

```
rape_cls = RandomForestRegressor()
```

```
rape_cls.fit(X, rape_Y)
```

```
calculateError(rape_cls, X_test2, y_test2)
```

```
murder_cls = RandomForestRegressor()
```

```
murder_cls.fit(X, murder_Y)
```

```
calculateError(murder_cls, X_test3, y_test3)
```

```
dataset = pd.read_csv("Dataset/Dataset.csv")

dataset.fillna(0, inplace = True)

columns = list(dataset.columns)

strdata = '<table border=1 align=center width=100%><tr><th><font size=""
color="black">' + columns[0] + '</th>'

for i in range(1, len(columns)):

    strdata += '<th><font size="" color="black">' + columns[i] + '</th>'

strdata += "</tr>"

dataset = dataset.values

for i in range(len(dataset)):

    strdata += "<tr>"

    for j in range(len(dataset[i])):

        strdata += '<td><font size="" color="black">' + str(dataset[i, j]) + '</td>'

    strdata += "</tr>"

context = {'data': strdata}

return render(request, 'ViewDataset.html', context)
```